



Audio Engineering Society

Convention e-Brief 129

Presented at the 145th Convention
2018 October 17 – 20, New York, NY, USA

This Engineering Brief was selected on the basis of a submitted synopsis. The author is solely responsible for its presentation, and the AES takes no responsibility for the contents. All rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Audio Engineering Society.

Introducing a dataset of guitar amplifier sounds for nonlinear emulation benchmarking

Thomas Schmitz¹ and Jean-Jacques Embrechts¹

¹University of Liège, Montefiore Institute, Liège, Belgium

Correspondence should be addressed to Thomas Schmitz (T.Schmitz@uliege.be)

ABSTRACT

Recent progresses made in the nonlinear system identification field have improved the ability to emulate nonlinear audio systems such as tube guitar amplifiers. As a straightforward comparison of different models cannot always be made, we propose a new reference dataset enabling to highlight the strengths and weaknesses of different nonlinear modeling methods. Our dataset gathers five different styles of guitar sounds passing through different guitar amplifiers with 10 steps of their gain parameter (i.e. the distortion level of the amplifier). Moreover, a Matlab function is also provided to obtain the desired input/output sounds in a matrix form.

INTRODUCTION

As most real-world devices exhibit nonlinear behaviors, the modeling of nonlinear systems has been a central topic in many engineering areas. This project focuses on a main application: the modeling of guitar amplifiers. The idea is to replace these devices by a computer emulation that could be suitable for guitar players in many situations. Recent improvements enable to emulate a tube amplifier in real time with a good accuracy [1, 2]. Applied to our previous dataset, state-of-the-art methods reach a Root Mean Square Error (RMSE) of less than 2% between the real sound and the emulated one. The next challenge lies in the comparison of different Performance Indexes (PI) evaluated on several models. For example, these indexes could characterize the model in terms of accuracy, computational time, robustness, etc. To do so, a new dataset [3] composed of different types of guitar signals has been created. For creating this dataset, the guitar signals have been

played through several amplifiers. For each of these amplifiers, their output signal has been recorded for ten positions of the *gain* parameter where 1 is the lowest level of distortion and 10 the strongest one.

EXPERIMENTAL SETUP

Five different types of music have been recorded using a guitar *PRS SE 245* and the *UR22 Steinberg* sound card (44100hz sample frequency and float 32 bit depth). The 5 resulting musical samples have been concatenated in a *wav* file with a silence of two seconds inserted between them. Then, this *wav* file is sent through several guitar amplifiers (currently : the Ibanez *TSA15* channel *Crunch*, the Engl *Retro Tube50* channel *Drive*, the Mesa *Boogie 5-50+* channels *Clean*, *Crunch* and *Burn*, the Mesa *Boogie MarkV* channels *Clean*, *Crunch*, *Xtreme* and the *Blackstar HT5 Metal* channel *Disto*). The signal is then caught at the output of the power amplifier stage by a load box (*the Torpedo Reload* [4])

reducing its amplitude to a line level. Finally, the load box sent the signal back to the sound card (as presented in Fig. 1). A description of the 5 musical styles played for each sample is given hereafter.

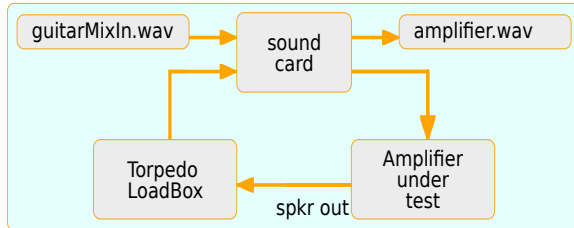


Fig. 1: Configuration settings for the recording of the dataset, where *amplifier.wav* is a scaled version of the recorded signal taken at the output of the amplifier (spkr out) when a guitar signal is applied at the input.

Sample1: chromatic scale

In this sample, each note of the PRS guitar neck is played successively. Starting on the 6th string of the guitar (E low), the notes are played until the 4th fret of the guitar neck. Then, the same pattern is applied from the 5th to the 1st string. When the 1st string is reached, all the fingers are shifted one fret to the right (as presented in Fig. 2). This sample ends when the 22th fret of the guitar neck is reached. One can notice that a *tablature* is provided along with the partition. A tablature depicts the six strings of the guitar. A number i corresponds to the i^{th} fret (see [5] to get further information).

Sample2: chords

A set of chords has been played in this order: C, D, E, F, G, A, B in Major then in Minor. The chords are given in Fig. 3.

Sample3: Bamba song

This sample is composed of a song of three chords: C, F, G as presented in Fig. 4.

Sample4: blues song

This sample is composed of a classical blues pattern: E, A, B7 as presented in Fig. 5.

Sample5: blues scale Am

This sample is a short solo of guitar in Am blues scale as presented in Fig.6.

BUILDING A LEARNING DATASET

For each amplifier, our five musical samples are sent to the guitar amplifier and recorded for ten different positions of the *Gain* parameter. Consequently, for 3 amplifiers with 3 channels, 90 recorded samples are produced. A Matlab function is provided to help the user to build its *learning set* (train+test sets) easily:

$$[train, test] = getInOut(modelName, lStyle, lGain, R)$$

This function enables the user to choose which sample type, amplifier and gain will be in its learning set and returns two matrices (*train* and *test*) composed of the input sound in the first column, its corresponding output sound in the second column and the gain parameter used during the recording in the third one. The parameters of the function are :

- *modelName*: the name of the *wav* file containing the output sound from one amplifier
- *lStyle*: a list of numbers corresponding to the desired musical samples among the ones presented in the previous section.
- *lGain*: a list of *Gain* parameter values.
- *R*: the ratio of the length of the *training* to the *test* sets.

Example:

$$[train, test] = getInOut('Engl.wav', [5, 3], [1, 10], 0.8)$$

returns two matrices containing the concatenation of the sample 5 and 3 with the gain parameter 1 and 10 as it can be seen in Fig. 7. The *train* matrix is used to train the model while the *test* matrix is used to validate the model on a sequence that has never been seen before. It ensures that the model does not over-fit the dataset. By default a used sample is sliced as follows: 80% of it is put in the *training* set while the remaining 20% are put in the *test* set. The slices are chosen randomly under the constraint that they remain the same each time the user call the function *getInOut*.



Fig. 2: Chromatic scale: all the notes of the guitar are played successively (sample1)

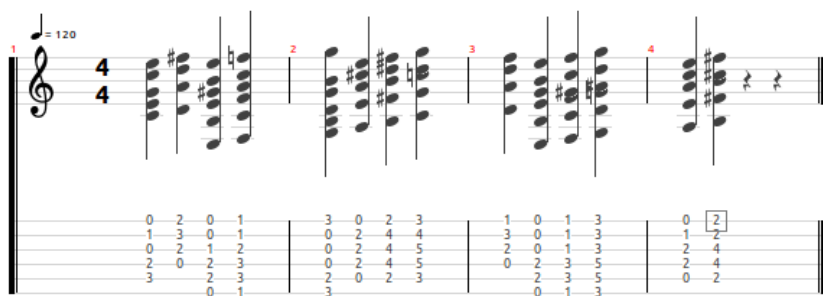


Fig. 3: Chord in Major scale then in Minor scale (sample2)

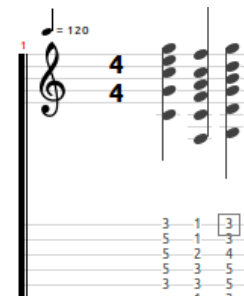


Fig. 4: Chord the song *La Bamba* (sample3)

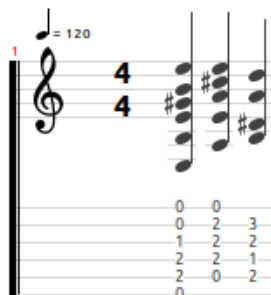


Fig. 5: Chord the blues song (sample4)



Fig. 6: Am blues scale (sample5)

Matrix train

C1	C2	C3
sample3[1]	out3[1]	1
⋮	⋮	⋮
sample3[N]	out3[N]	1
sample5	out5	1
sample3	out3	10
sample5	out5	10

Fig. 7: The matrices *Train* or *Test* contain the desired input samples in the first column (C1), their corresponding output signals from the amplifier in the second column (C2) and the gain parameter value used during the recording in the third column (C3).

COMPARISON METHODOLOGY

The comparison of two models in term of accuracy can be reported as a Performance Index (PI) that measures the distance between the output of the amplifier y_{amp} and the output of the model y_{model} :

$$PI = \sqrt{\frac{1}{K} \sum_{k=0}^K [y_{amp}[k] - y_{model}[k]]^2}, \quad (1)$$

where K is the number of samples of the signal y . This PI should be measured on the test set and not on the training set to be sure that the model generalize well. The users of the dataset can also specify an other PI formula that is better suited to their needs. The Table. 1 presents the PI (Root Mean Square Error) when the method described in [1] is applied to the dataset. For this evaluation, we used all the guitar samples and we set the gain of each amplifier on the middle position: 5.

CONCLUSION

In this paper we propose a new dataset to compare different nonlinear methods applied, in this case, to the emulation of guitar amplifiers. We believe that this dataset will help researchers to find out the strengths, weaknesses, limitations and reliability of their model. We hope that every one will see it as a way to better understand the nonlinear real world rather a competition between researchers. With this aim in mind, we

Table 1: Example of Performance Index:

Amplifier	Sample style	Gain	PI
Engl retro tube disto	1,2,3,4,5	5	6%
MesaBoogie 550 clean	1,2,3,4,5	5	10%
MesaBoogie 550 crunch	1,2,3,4,5	5	9%
MesaBoogie 550 disto	1,2,3,4,5	5	11%
Ibanez TSA15 crunch	1,2,3,4,5	5	7%
MesaBoogie MarkV clean	1,2,3,4,5	5	1%
MesaBoogie MarkV crunch	1,2,3,4,5	5	6%
MesaBoogie MarkV disto	1,2,3,4,5	5	9%
Blackstar HT5M	1,2,3,4,5	5	9%

Note: The implementation of method [1] enables to use the model in real time on a Nvidia Titan Xp. In practice, even with an error of 10% the difference is inaudible.

have started to share the codes of our methods since we believe that this kind of interactions can push the nonlinear system modeling field forward. Table 1 can be considered as an example of description of results for this dataset.

References

- [1] Schmitz, T. and Embrechts, J.-J., “Nonlinear Real-Time Emulation of a Tube Amplifier with a Long Short Term Memory Neural-Network,” in *Presented at the 144th Convention of the Audio Engineering Society*, May 2018.
- [2] Schmitz, T. and Embrechts, J.-J., “Real Time Emulation of Parametric Guitar Tubes Amplifier With Long Short Term Memory Neural Network,” in *5th International Conference on Signal Processing (CSIP2018)*, March 2018.
- [3] Schmitz, T., “Dataset of guitar amplifiers for emulation purpose,” <http://www.montefiore.ulg.ac.be/services/acous/STSI/downloads.php>, 2018.
- [4] TwoNoteEngineering, “Torpedo Reload,” <https://www.two-notes.com/fr/torpedo-reload>, 2011.
- [5] Wikipedia, “How to read tablatures,” <https://en.wikipedia.org/wiki/Tablature>, 2018.